

PEMBANGUNAN APLIKASI MANAJEMEN ASET

by Hariyady Hariyady

Submission date: 14-Jan-2020 10:17AM (UTC+0700)

Submission ID: 1241699463

File name: 2471-6213-1-PB.pdf (967.58K)

Word count: 4192

Character count: 28482

PEMBANGUNAN APLIKASI MANAJEMEN ASET “MYASSET” BERBASIS MOBILE MENGGUNAKAN METODE INCREMENTAL

Hariyady

Teknik Informatika, Universitas Muhammadiyah Malang, Malang

Kontak Person:

Hariyady

Universitas Muhammadiyah Malang

E-mail: hariyady@umm.ac.id

Abstrak

Manajemen aset merupakan faktor internal perusahaan atau individu dalam mengelola sumber daya secara aktif dan efisien. Pengelolaan sumberdaya yang efektif dan efisien akan mengotimalkan operasi dari perusahaan atau individu dan meningkatkan keunggulan kompetitif. Perencanaan dan implementasi teknologi informasi yang baik akan menjadi adalan utama dalam mengelola sumberdaya yang optimal. Perkembangan aset juga membutuhkan perangkat lunak manajemen pengelolaan aset yang baik. Perangkat lunak manajemen aset yang ada pada umumnya fokus untuk menangani pengelolaan aset yang berkaitan dengan jumlah secara kuantitas. Sedangkan sebaran aset kurang mendapat perhatian sehingga menimbulkan kendala dalam melakukan peremajaan data. Kedala dalam peremajaan data akan mengakibatkan inkonsistensi data yang menimbulkan kurang optimalnya pengelolaan aset. Model proses pengembangan perangkat lunak yang digunakan adalah incremental/bertahap. Model proses incremental digunakan karena berkaitan dengan karakteristik aplikasi yang dibangun yang terdiri dari beberapa sub sistem. Pembangunan aplikasi MyAsset dikembangkan dengan menggunakan model proses incremental. Pengembangan model proses incremental dipilih untuk pengembangan perangkat lunak secara bertahap. Setiap siklus pengembangan akan dibangun mulai dari fitur yang memiliki skala prioritas yang dibutuhkan pengguna. pengembangan incremental adalah manajemen pengembangan yang sederhana. Pengembangan perangkat lunak dilakukan bertahap dan diverifikasi ke pengguna sehingga pengembangannya lebih terkontrol. Siklus pengembangan dimulai dari fitur utama aplikasi yang menjadi prioritas utama pengguna. Setiap tahap pengembangan perangkat lunak yang sudah dilakukan pengujian dapat digunakan langsung oleh pengguna tanpa harus menunggu semua tahapan selesai. Hal ini mengurangi kegagalan pengembangan proyek perangkat lunak karena siklus pengembangan dilakukan verifikasi kepada pengguna.

Kata kunci: Sistem Manajemen Aset, Model Proses Incremental, Aplikasi Aset Mobile

1. Pendahuluan

Pada era globalisasi, perusahaan dan individu harus memiliki keunggulan kompetitif untuk memenangkan persaingan bisnis. Teknologi informasi merupakan keunggulan kompetitif yang diyakini sebagai faktor internal perusahaan dalam mengelola sumberdaya secara efektif dan efisien. Pengelolaan sumberdaya yang efektif dan efisien akan mengotimalkan operasi dari perusahaan atau individu dan meningkatkan keunggulan kopetitif. Perencanaan dan implementasi teknologi informasi yang baik akan menjadi adalan utama dalam mengelola sumberdaya yang optimal.

Perkembangan perusahaan atau individu berdampak pada perkembangan aset yang dimiliki perusahaan atau individu. Perkembangan aset menyangkut jumlah secara kuantitas dan distribusi aset yang dimiliki perusahaan atau individu. Perkembangan aset juga membutuhkan perangkat lunak manajemen pengelolaan aset yang baik. Perangkat lunak manajemen aset yang ada pada umumnya fokus untuk menangani pengelolaan aset yang berkaitan dengan jumlah secara kuantitas. Sedangkan sebaran aset kurang mendapat perhatian sehingga menimbulkan kendala dalam melakukan peremajaan data. Kedala dalam peremajaan data akan mengakibatkan inkonsistensi data yang menimbulkan kurang optimalnya pengelolaan aset.

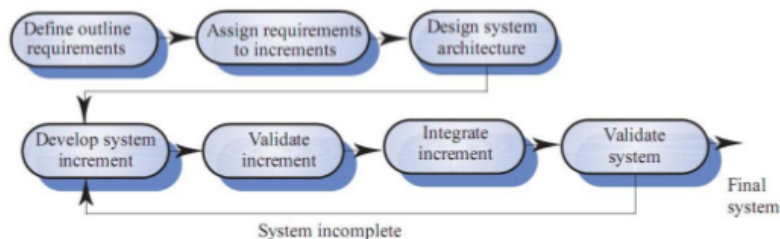
Perangkat lunak manajemen aset yang berkembang saat ini masih terbatas pada proses dokumentasi. Sedangkan kebutuhan perangkat lunak manajemen aset tidak terbatas pada dokumentasi namun merupakan proses terintegrasi yang meliputi permintaan aset, persetujuan permintaan, pembelian, register aset, pengiriman aset, dan penerimaan aset [1]. Proses permintaan aset masih menggunakan dokumen yang harus ditandatangani manajemen dan data aset yang masih disimpan dalam bentuk file. Hal ini kurang mengakomodasi kepentingan perusahaan dalam penelusuran permintaan, realisasi dan administrasi aset. Kompleksitas manajemen aset membutuhkan rancangan perangkat lunak yang dapat memenuhi kebutuhan integrasi yang melibatkan banyak proses dan aktor.

Dewasa ini, perkembangan teknologi *mobile* telah membawa dampak yang sangat besar dalam kehidupan. Perkembangan teknologi *mobile* secara perangkat keras memiliki ukuran dan bentuk yang sangat menarik. Sedangkan secara perangkat lunak tersedia banyak aplikasi yang mendukung perangkat *mobile* sehingga fungsi dari perangkat *mobile* tidak hanya untuk sebatas telpon dan sms aja. Perkembangan perangkat lunak teknologi *mobile* membawa dampak terhadap aktifitas bisnis. Dengan demikian, aktifitas bisnis dapat dilakukan dimanapun tanpa tergantung dimensi jarak dan waktu. Faktor fleksibilitas dan portabilitas dalam aktifitas bisnis merupakan faktor penting yang menjadi keunggulan dari aplikasi berbasis *mobile*.

Pengembangan perangkat lunak manajemen aset berbasis *mobile* diharapkan dapat menjadi solusi permasalahan di atas. Permasalahan peremajaan data dapat diatasi dengan membangun perangkat lunak yang memiliki portabilitas dan fleksibilitas yang baik. Sehingga peremajaan data dapat dilakukan dengan mudah melalui perangkat *mobile* tanpa harus menggunakan terminal komputer. Hal ini mendukung kondisi keberadaan aset yang tersebar pada lokasi yang berbeda-beda. Teknologi *mobile* juga dapat memberikan solusi permasalahan interaksi yang melibatkan operasi antar prosedur dan melibatkan banyak pihak pemangku kepentingan. Pihak pemangku kepentingan dapat memberi persetujuan permintaan aset dari perangkat *mobile*, lebih jauh penelusuran dan pemantauan aset dapat dilakukan lebih mudah dengan dukungan API QR-code.

2. Metode Penelitian

Model proses pengembangan perangkat lunak yang digunakan adalah *incremental*/bertahap. Model proses *incremental* digunakan karena berkaitan dengan karakteristik aplikasi yang dibangun yang terdiri dari beberapa sub sistem. Pada model proses *incremental* kebutuhan pengguna diprioritaskan dan prioritas tertinggi dimasukan *incremental* awal. Pada saat pengembangan *increment* dimulai maka kebutuhan dibekukan dahulu sampai tahap *increment* berikutnya dimulai. Sedangkan keuntungan penggunaan model proses *incremental* adalah (1) nilai penggunaan dapat ditentukan setiap awal *increment* sehingga fungsionalitas sistem dapat disediakan sejak awal, (2) *increment* awal berupa *prototype* untuk membantu memahami kebutuhan pada *increment* berikutnya dan (3) Memiliki risiko lebih rendah terhadap keseluruhan pengembangan sistem. Adapun tahapan pengembangan sistem model proses *incremental* seperti pada Gambar 1. Sedangkan Gambar 2 menjelaskan rancangan metodologi penelitian dengan mengacu pada model proses *incremental*.



Gambar 1 Model Proses Incremental [8]

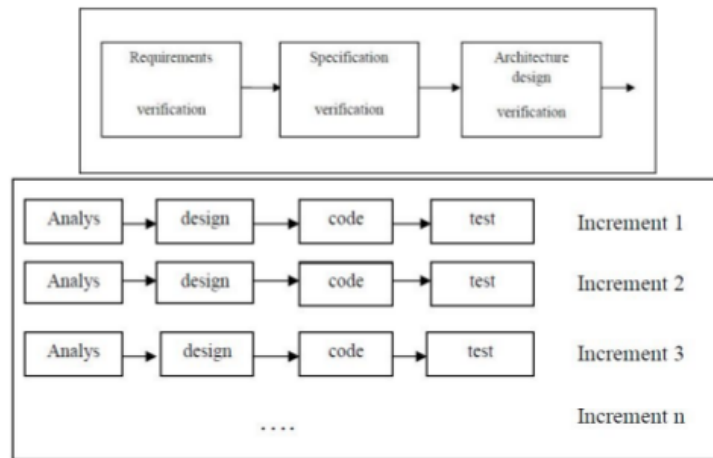
Alur metodologi penelitian terdiri dimulai dengan memahami *domain* permasalahan di bidang manajemen Aset. Pemahaman *domain* permasalahan dilakukan dengan melakukan studi literatur. Selanjutnya dilakukan elisitasi dan analisis kebutuhan sistem. Pada tahap ini akan menghasilkan kebutuhan fungsional dan kebutuhan non fungsional dari sistem. Pada tahap perancangan dilakukan perancangan pada basis data dan sistem. Perancangan basis data dirancang dengan menggunakan ER Diagram sedangkan perancangan sistem dilakukan dengan menggunakan beberapa diagram UML antara lain: *sequence diagram*, *activity diagram*, *class diagram* dan desain antarmuka. Hasil rancangan akan diimplementasikan kedalam lingkungan pengembangan. Implementasi lingkungan pengembangan aplikasi menggunakan Android Studio sedangkan basis data menggunakan MySql pada *server* dan MySql lite pada *device mobile*. Teknik pengujian difokuskan untuk menguji secara fungsional setiap fitur sehingga digunakan teknik *black box*. Siklus pengembangan sistem akan dilakukan iterasi dengan menggunakan model proses *incremental* / bertahap dari fitur utama. Alur proses penelitian sebagaimana tampak pada gambar 2.

Metode pengembangan perangkat lunak *incremental* menekankan pada pembagian *requirement* kedalam beberapa fungsi atau bagian sehingga *model* pengembangannya dilakukan secara bertahap. Tahapan pengembangan Incremental terdiri dari: (1) *requirement*, (2) *specification*, (3) *architecture design*, (4) *code* dan (5) *test*. *Requirement* adalah proses tahapan awal pengembangan perangkat lunak, yaitu menentukan kebutuhan atau analisa kebutuhan yang menjadi basis iterasi. *Specification* adalah tahapan spesifikasi perangkat lunak mengacu pada tahap analisa kebutuhan. *Architecture design* adalah proses perancangan perangkat lunak agar pembangunannya dapat dilakukan per bagian. *Code* adalah proses implemetasi ke lingkungan pengembangan hasil perancangan. Dan *Test* adalah proses pengujian setiap siklus iterasi pada model *incremental*. Pada metode *incremental* setiap tahapan dilakukan secara berurutan mulai dari *requirement* sampai test. Setiap tahap siklus (*increment*) pengembangan yang telah melalui proses pengujian dapat digunakan langsung oleh pengguna. Pengembangan perangkat lunak berlanjut ke siklus pengembangan berikutnya semua tahapan *increment* selesai dilakukan seperti Gambar 3.



Gambar 2 Diagram Alir Metodologi Penelitian

Kelebihan dan metode pengembangan *incremental* adalah manajemen pengembangan yang sederhana. Pengembangan perangkat lunak dilakukan bertahap dan diverifikasi ke pengguna sehingga pengembangannya lebih terkontrol. Siklus pengembangan dimulai dari fitur utama aplikasi yang menjadi prioritas utama pengguna. Setiap tahap pengembangan perangkat lunak yang sudah dilakukan pengujian dapat digunakan langsung oleh pengguna tanpa harus menunggu semua tahapan selesai. Hal ini mengurangi kegagalan pengembangan proyek perangkat lunak karena siklus pengembangan dilakukan verifikasi kepada pengguna.



Gambar 3 Model proses pengembangan Incremental

3. Hasil dan Pembahasan

Pembangunan aplikasi “MyAsset” dikembangkan dengan menggunakan model proses *incremental*. Pengembangan model proses *incremental* dipilih untuk pengembangan perangkat lunak secara bertahap. Setiap siklus pengembangan akan dibangun mulai dari fitur yang memiliki skala prioritas yang butuhkan pengguna. Pada siklus selanjutnya akan dikembangkan fitur-fitur penunjang untuk mendukung aplikasi MyAsset. Adapun proses pembangunan setiap tahap diuraikan berikut:

3.1 Studi Literatur

Siklus Manajemen aset merupakan proses pengelolaan aset mulai dari perencanaan, pengadaan, pemakaian dan penghapusan aset. Siklus manajemen aset mempertimbangkan semua pilihan dan strategi manajemen sebagai bagian dari masa pakai aset, perencanaan sampai penghapusan aset. Sedangkan tujuan adalah untuk mencari biaya terendah dalam jangka panjang (bukan penghematan dalam jangka pendek) ketika membuat keputusan dalam aset manajemen.



Gambar 4 Siklus Manajemen Aset [6]

Pada gambar 4 menggambarkan siklus manajemen aset yang melibatkan beberapa proses. Proses perencanaan aset meliputi konfirmasi tentang pelayanan yang dibutuhkan oleh pelanggan dan memastikan bahwa aset yang diajukan merupakan solusi yang paling efektif untuk memenuhi kebutuhan pelanggan. Proses pengadaan aset merupakan peningkatan dari aset dimana pembiayaan dapat menjadi alasan yang diharapkan untuk menyediakan keuntungan diluar tahun pembiayaan.

Pengoperasian aset mempunyai fungsi yang berhubungan dengan kerja, pengendalian aset dan biaya terkait merupakan komponen penting dalam aset yang dinamis atau berumur pendek. Penghapusan aset adalah pilihan ketika sebuah aset tidak diperlukan lagi, menjadi tidak ekonomis untuk dirawat atau direhabilitasi.

3.2 Tahap Analisa

Pada analisa dilakukan pembangunan perangkat lunak difokuskan untuk melakukan elisitasi terhadap kebutuhan perangkat lunak. Kebutuhan perangkat lunak menjadi faktor penting bagi pembangunan perangkat lunak. Hasil elisitasi kebutuhan meliputi: 1. permintaan aset, 2. persetujuan permintaan, 3. pembelian, 4. registrasi aset, 5. pengiriman aset, dan 6. penerimaan aset. Selanjutnya hasil elisitasi dilakukan analisa kebutuhan untuk memperoleh kebutuhan fungsional dari sistem. Hasil analisa kebutuhan akan mendeskripsikan kebutuhan fungsional dari sistem seperti tampak pada Tabel 1 dan kebutuhan non fungsional tampak seperti Tabel 2.

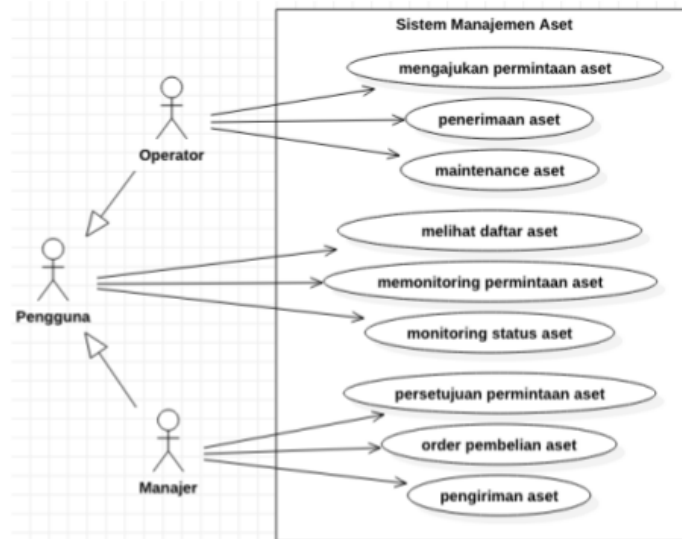
Tabel 1 Kebutuhan fungsional

No	Deskripsi SRS
A-SRS-F-01	Sistem menyediakan fasilitas melihat daftar aset
A-SRS-F-02	Sistem menyediakan fasilitas mengajukan permintaan aset
A-SRS-F-03	Sistem menyediakan fasilitas memonitoring pengajuan permintaan aset
A-SRS-F-04	Sistem menyediakan fasilitas persetujuan permintaan aset
A-SRS-F-05	Sistem menyediakan fasilitas <i>order</i> pembelian permintaan aset yang telah disetujui
A-SRS-F-06	Sistem menyediakan fasilitas pengiriman aset ke unit kerja
A-SRS-F-07	Sistem menyediakan fasilitas penerimaan aset
A-SRS-F-08	Sistem menyediakan fasilitas <i>maintenance</i> aset
A-SRS-F-09	Sistem menyediakan fasilitas untuk memonitoring status aset

Tabel 2 Kebutuhan non fungsional

No	Deskripsi SRS
A-SRS-NF-01	Sistem dilengkapi dengan fasilitas <i>login</i> untuk keperluan autentikasi pengguna
A-SRS-NF-02	Sistem dapat dijalankan minimal menggunakan OS android jelly bean
A-SRS-NF-03	Sistem memiliki waktu respon yang cepat

Hasil analisa kebutuhan berupa kebutuhan fungsional akan diabstraksikan dalam bentuk *use case diagram*. Abstrasi kedalam *use case diagram* untuk meningkatkan pemahaman pengguna terhadap fitur aplikasi. Pemahaman terhadap fitur aplikasi akan meningkatkan kualitas ada taham desain. *use case diagram* yang telah tersusun sebagaimana terlihat pada Gambar 5. Pada *use case diagram* terdapat 3 aktor yaitu: aktor pengguna, aktor operator dan aktor manajer. Aktor pengguna merupakan aktor generalisasi dari aktor operator dan aktor manajer. Aktor pengguna memiliki 3 case yaitu: 1. Melihat daftar aset, 2. Memonitoring permintaan aset dan 3. memonitoring status aset. Ketiga *use case* tersebut akan dapat diakses oleh aktor operator dan manajer dengan *scope* yang berbeda. Perbedaanya Aktor operator hanya dapat mengakses 3 case sesuai dengan *scope* unit kerja dari aktor operator sedangkan aktor manajer dapat mengakses ketiga case tersebut tampa dibatasi *scope* unit kerja. Pada aktor operator memiliki 3 case spesifik, yaitu 1. Mengajukan permintaan aset, 2. Penerimaan aset dan 3. *Maintenance* aset. Sedangkan pada aktor manajer terdapat 3 *use case* spesifik yaitu: 1. Persetujuan permintaan aset, 2. *Order* pembelian aset dan pengiriman aset. Integrasi pada masing-masing case akan menyusun sistem manajemen akses yang terintegrasi merujuk pada siklus manajemen aset.



Gambar 5 Use case diagram sistem manajemen aset

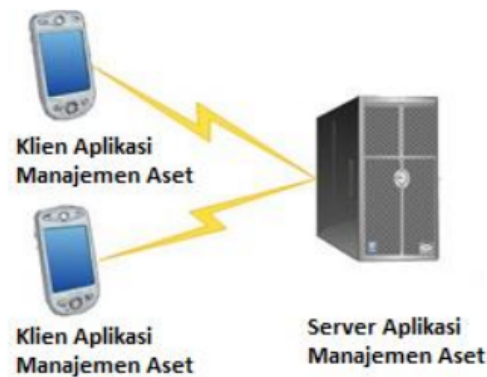
Penggunaan model proses *Incremental* akan membagi semua case kedalam 2 *increment*. *Increment* yang pertama terdiri dari case mengajukan permintaan aset, melihat daftar aset, memonitoring permintaan aset dan persetujuan permintaan aset. Pada *increment* pertama akan menghasilkan *prototype* pertama dari sistem manajemen aset. Tahap berikutnya akan menambahkan *use case* : penerimaan akan aset, *maintenance aset*, *monitoring status aset*, order pembelian dan pengiriman aset kedalam *increment* kedua. Pada tahap *increment* yang kedua sistem akan menghasilkan *prototype* sistem manajemen aset yang memiliki fitur yang lengkap. Skema pembagian *use case* dalam model proses *Incremental* seperti tampak pada Tabel 3.

Tabel 3 Skema pembagian *use case* pada model proses *Incremental*

No	Use case	Increment
1.	Mengajukan permintaan aset	Increment 1 → Prototype pertama
2.	Melihat daftar aset	
3.	Memonitoring permintaan aset	
4.	Persetujuan permintaan aset	
5.	Penerimaan akan aset	
6.	Maintenance aset	Increment 2 → Prototype kedua
7.	Monitoring status aset	
8.	Order pembelian	
9.	Pengiriman aset	

3.3 Tahap Desain

Melakukan perancangan terhadap perangkat lunak yang akan dikembangkan berdasarkan hasil yang diperoleh dari analisa. Secara umum Perancangan arsitektur aplikasi seperti pada Gambar 6. Arsitektur aplikasi menggunakan arsitektur *client server*, *server* akan menyimpan semua data aset. Sedangkan *client* menggunakan pendekatan pengembangan aplikasi *smart client*. Penggunaan arsitektur ini *client* akan mengakses data pada *server* dan menyimpan data *client* pada *database local* pada perangkat *mobile*. Keunggulan dari arsitektur *smart client* akan memperingan beban *server*, baik dari sumberdaya sistem komputasi dan jaringan. *Client* tidak akan mengakses data pada *server* jika *database client* sudah tersinkronisasi namun cukup mengakses *database local* pada *client*. Jika terjadi permasalahan akses ke *server* berkaitan dengan permasalahan *resource server* dan jaringan, *client* masih dapat menjalankan operasinya dengan mengakses *database local*.



Gambar 6 Rancangan Umum Aplikasi Sistem Manajemen Aset [1-4]

Merujuk model proses *Incremental* pada pengembangan aplikasi sistem manajemen aset maka desain *user interface* terbagi menjadi menjadi 2 tahap yaitu: tahap desain *user interface* pada *prototype* 1 dan 2. Pada *prototype* 1 akan didesain *user interface* untuk case: mengajukan pengadaan aset, melihat daftar aset, memonitoring permintaan aset dan persetujuan permintaan aset.



Gambar 7 Mockup use case mengajukan permintaan aset dan melihat daftar aset

Pada Gambar 7 menunjukkan *mockup* untuk *use case* mengajukan permintaan aset dan melihat daftar aset. Pada *mockup* mengajukan pengadaan aset terdapat elemen *user interface*: kode pengajuan yaitu kode yang dibangkitkan secara otomatis oleh sistem. Kode pengajuan bersifat unik yang bertujuan untuk mengidentifikasi daftar pengajuan pengadaan barang. Selanjutnya tanggal pengajuan adalah tanggal pengajuan pengadaan aset, tanggal pengajuan untuk mengidentifikasi tanggal pengajuan pengadaan aset dan menjadi basis perhitungan lamanya pengadaan aset. Selanjutnya *user interface* yang berkaitan dengan atribut aset yang akan diajukan. Atribut aset terdiri dari: id aset, nama aset, jumlah, satuan, harga satuan, total dan penempatan aset. pengisian atribut dari aset akan ditampilkan dalam daftar (*list*) pengadaan aset. Daftar pengadaan aset akan menampilkan sejumlah *record* aset yang diajukan oleh suatu unit kerja pada periode pengajuan aset. daftar pengadaan aset dilengkapi dengan tiga aksi yaitu tombol untuk menambahkan data aset kedalam daftar aset. Tombol untuk menghapus data aset dari daftar aset dan fasilitas pencarian untuk mencari data dari *list* aset. Tombol pengajuan aset akan memproses pengajuan daftar aset yang sudah terdaftar dalam daftar aset.

Pada mockup melihat aset terdapat elemen *user interface*: *user interface* yang terdiri atribut-atribut aset dan *list* aset. Daftar aset menampung data aset yang ada dan dilengkapi fitur navigasi (◀ ▶ ⌂), fitur navigasi membantu pengguna untuk menelusuri *record* dalam daftar aset. Disamping tombol navigasi daftar aset juga dilengkapi dengan fitur pencarian untuk memudahkan fitur pencarian. Setiap *record* yang ditunjuk pada *list* aset, data secara detil akan ditampilkan dalam form aset yang memiliki atribut aset terdiri dari: id aset, nama aset, jumlah, satuan, harga satuan, total dan penempatan aset.



Gambar 9 Mockup use case memonitoring permintaan aset dan persetujuan permintaan aset

Pada Gambar 8 menunjukan *mockup* untuk *use case* memonitoring permintaan aset dan persetujuan permintaan aset. Pada desain *interface* memonitoring permintaan aset bertujuan untuk memonitoring pengajuan aset yang diajukan pengguna. Atribut kode pengajuan dan tanggal pengajuan adalah kunci untuk menampilkan daftar aset. Setiap daftar aset yang ditampilkan akan terlihat statusnya: diproses, disetujui, diorder, dikirim. Daftar pengajuan aset dilengkapi dengan fitur navigasi (◀ ▶ ⌂) untuk memudahkan pengguna menelusuri data aset yang diajukan. Setiap data aset yang ditunjuk akan ditampilkan dalam form aset untuk melihat lebih detil atribut-atribut dari aset. disamping itu daftar pengajuan aset juga dilengkapi fitur pencarian (🔍 cari ...). fitur pencarian sangat membantu jika data dalam daftar pengajuan aset memiliki jumlah *record* yang cukup banyak.

Mockup use case persetujuan permintaan aset merupakan desain *user interface* yang dirancang untuk aktor manajer untuk memberikan persetujuan terhadap permintaan aset. Atribut kode pengajuan dan tanggal pengajuan menjadi kunci data pengajuan aset yang akan ditampilkan dalam daftar pengajuan aset. daftar pengajuan aset memiliki fitur navigasi (◀ ▶ ⌂) dan fitur pencarian (🔍 cari ...). kedua fitur bertujuan untuk memberikan kemudahan pengguna menelusuri data. Data yang sedang ditunjuk akan ditampilkan kedalam *form* aset untuk memudahkan pengguna meng-*update* data persetujuan.

Selanjutnya akan dilakukan pembangunan *prototype* kedua dengan mengesekusi *increment* kedua dari model proses *Incremental*. Pada dasarnya *increment* 2 adalah menambahkan fitur-fitur tambahan kedalam *prototype* pertama yang merupakan produk dari *increment* pertama. Adapun fitur tambahan pada *increment* kedua adalah : Penerimaan akan aset, *Maintenance* aset, Monitoring status aset, *Order* pembelian dan Pengiriman aset. pada *increment* kedua akan menghasilkan produk *prototype* kedua yang memiliki fitur yang lebih lengkap. secara teoritis, pengembangan fitur dengan menggunakan model proses *incremental* masih dapat ditambahkan terus.

Pada Gambar 9 menggambarkan *use case* penerimaan aset dan *maintenance* aset. Penerimaan aset merupakan fitur yang dirancang proses penerimaan aset oleh unit kerja. Fitur penerimaan aset

merupakan finalisasi rangkaian proses pengajuan permintaan barang, persetujuan permintaan aset, order pembelian aset dan pengiriman aset. Basis operasi dari fitur ini adalah masukan kode pengajuan dan tanggal pengajuan. Selanjutnya list penerimaan aset akan menampilkan data sesuai dengan masukan kode pengajuan. Pengguna dapat menggunakan tombol navigasi < > <> untuk menelusuri data penerimaan barang. Ketika data ditunjuk data detail akan ditampilkan kedalam form aset yang memudahkan pengguna untuk mengupdate data kondisi aset yang terima.

Sedangkan fitur *maintenance aset* adalah fitur bagi aktor operator aplikasi manajemen aset untuk megupdate data *maintenance*. Fitur *maintenance aset* berbasis tanggal *maintenance* dan id aset yang menjadi masukan. Masukan id aset akan menampilkan atribut aset secara lengkap dan selanjutnya pengguna memasukkan atribut jenis dan biaya *maintenance* pada saat aset dilakukan *maintenance*. Tombol + untuk menambahkan data aset kedalam *list maintenance* dan Tombol - untuk menghapus data aset dari *list maintenance*.



Gambar 9 Mockup use case penerimaan aset dan maintenance aset

Pada Gambar 10 merupakan *Mockup use case monitoring status aset dan order pembelian aset*. fitur monitor aset digunakan oleh aktor pengguna yang merupakan generalisasi dari aktor operator dan manajer sehingga fitur ini dapat dijalankan oleh aktor operator dan manajer. Basis operasi dari fitur ini adalah kode pengajuan, tanggal pengajuan dan id aset. Dengan masukan ketiga atribut tersebut data aset akan ditampilkan ke dalam *list* daftar *monitoring status aset*. pengguna dapat menelusuri data aset yang dalam dengan menggunakan tombol navigasi atau dengan < > <> atau dengan memasukan kunci pencarian pada fitur . Data yang ditunjuk akan ditampilkan pada form aset yang dilengkapi dengan atribut status aset sehingga pengguna dapat meng-update status aset dengan mudah.

Sedangkan fitur *order pembelian aset* adalah fitur lanjutan proses persetujuan permintaan aset. data persetujuan permintaan aset yang telah disetujui merupakan data yang digunakan pada *mockup* ini. fitur ini berbasis kode pengajuan dan tanggal pengajuan dalam operasinya. Data *order pembelian* akan ditampilkan dalam daftar *order pembelian aset*. Tombol *order aset* akan mengupdate tanggal pembelian aset sesuai dengan tanggal saat itu. Sedangkan tombol cetak SPK akan mencetak surat perintah kerja (SPK) berdasarkan daftar order pembelian barang. Tombol navigasi < > <> atau fitur pencarian akan memudahkan aktor manajer dalam menelusuri data *order pembelian*.

Pada Gambar 11 menunjukan *mockup use case pengiriman aset*. fitur pengiriman aset bertujuan untuk mengirimkan aset ke unit kerja yang mengajukan aset. fitur pengiriman aset merupakan fitur lanjutan dari fitur pengajuan permintaan aset, persetujuan permintaan aset dan *order pembelian aset*. Basis operasi dari fitur ini adalah masukan kode pengajuan dan tanggal pengajuan. Selanjutnya masukan akan menjadi kunci untuk menampilkan daftar pengiriman aset. Dalam daftar ini dilengkapi dengan tombol navigasi < > <> dan fasilitas pencarian untuk menelusuri

data aset. Data yang aktif pada daftar akan ditampilkan secara detil dalam *form* aset untuk memudahkan pengguna meng-*update* data. Tombol pengiriman aset berfungsi untuk memproses pengiriman barang kepada unit penerima. Sedangkan tombol cetak tanda terima berfungsi jika diperlukan dokumen tanda terima secara *hard copy*.



Gambar 10 Mockup use case monitoring status aset dan order pembelian aset



Gambar 11 Mockup use case pengiriman aset

3.4 Tahap Implementasi

Lingkungan pengembangan aplikasi "MyAsset" diimplementasikan menggunakan android studio dengan bahasa pemrograman Java. Android studio digunakan untuk mengimplementasikan rancangan (*mockup setiap case*) ke dalam lingkungan pengembangan *native android*. Android studio merupakan *Integrated Development Environment (IDE)* yang banyak dipilih oleh developer android. Kelengkapan *tools* pengembangan, tidak berbayar dan banyaknya dukungan komunitas menjadikan android studio sebagai pilihan utama *developer native android*. Bahasa pemrograman java digunakan untuk memplementasi *mockup setiap use case* yang menggunakan pendekatan pemrograman berorientasi

obyek. Pendekatan pemrograman berorientasi obyek sangat cocok diterapkan untuk pemrograman yang melibatkan obyek real di dunia nyata seperti pada sistem manajemen aset. Implementasi beberapa desain pattern dalam pembangunan aplikasi akan mempersingkat waktu pengembangan dan meningkatkan kualitas perangkat lunak.

Implementasi pembangunan aplikasi sesuaikan dengan jumlah *increment* pada model proses *incremental* yang digunakan. Metode *incremental* sangat fleksible diterapkan pada pembangunan aplikasi yang dikembangkan secara bertahap. Aplikasi yang memiliki *scope* yang besar tidak diimplementasikan secara langsung namun dibagi menjadi beberapa *increment*. Hal ini untuk mengurangi resiko kegagalan pengembangan aplikasi. Implementasi setiap tahap *increment* disesuaikan dengan kebutuhan yang menjadi skala prioritas pengguna. Setiap tahap implementasi pada sebuah *increment*, pengguna melakukan verifikasi. Hal ini untuk menjamin pengembangan aplikasi dapat terkontrol dengan baik oleh pengguna. Pada implementasi *increment* pertama dihasilkan *prototype* pertama yang memiliki 4 fitur core aplikasi manajemen aset. Selanjutnya pada *increment* yang kedua ditambahkan 5 fitur tambahan untuk melengkapi fitur yang sudah ada. Implementasi setiap *increment* berjalan secara independent, pada saat implementasi *increment* kedua. *Prototype* pertama dapat berjalan dengan baik tanpa terganggu implementasi *increment* kedua. Pada saat *increment* yang kedua selesai, fitur pada *prototype* pertama sudah tergabung dengan baik dengan fitur *increment* kedua. pengguna selanjutnya mengoperasikan *prototype* kedua tanpa ada kendala penggunaan fitur-fitur aplikasi sebelumnya.

3.5 Tahap Pengujian

Tahap pengujian perangkat lunak aplikasi MyAset menggunakan pengujian fungsional. Pengujian fungsional untuk menguji kesesuaian fitur tahap analisa, desain dan implementasi. Pengujian dilakukan pada setiap tahap *increment* pengembangan aplikasi mengikuti model proses *incremental*. Pada *increment* pertama pengujian fungsional dilakukan pada 4 fitur yaitu: 1. mengajukan permintaan aset, 2. melihat daftar aset, 3. memonitoring permintaan aset dan 4. persetujuan permintaan aset. Setelah melalui pengujian secara fungsional semua fitur *prototype* pertama, aplikasi siap dioperasikan oleh pengguna. Pada *increment* yang kedua, pengujian dilakukan pada 5 fitur yaitu: Penerimaan aset, maintenance aset, *monitoring* status aset, *order* pembelian aset dan pengiriman aset. Jika pengujian fungsional telah berjalan dengan baik maka *prototype* kedua yang memiliki fitur gabungan siklus pengembangan *increment* pertama dan kedua dapat dioperasikan oleh pengguna.

4. Kesimpulan

Pembangunan aplikasi “MyAsset” dikembangkan dengan menggunakan model proses *incremental*. Pengembangan model proses *incremental* dipilih untuk pengembangan perangkat lunak secara bertahap. Setiap tahap siklus pengembangan akan dibangun mulai dari fitur yang memiliki skala prioritas yang dibutuhkan pengguna. Pengembangan *incremental* adalah manajemen pengembangan yang sederhana. Pengembangan perangkat lunak dilakukan bertahap dan diverifikasi ke pengguna sehingga pengembangannya lebih terkontrol. Siklus pengembangan dimulai dari fitur utama aplikasi yang menjadi prioritas utama pengguna.

Pengembangan aplikasi manajemen aset “MyAsset” dibagi menjadi dua *increment*. Pembagian *increment* dengan mempertimbangkan prioritas fitur yang paling dibutuhkan pengguna. Jumlah fitur dalam *increment* dibuat seimbang karena pertimbangan waktu pengembangan yang hampir sama pada setiap *increment*. Pada *increment* pertama dihasilkan *prototype* pertama yang memiliki 4 fitur yaitu: mengajukan permintaan aset, melihat daftar aset, memonitoring permintaan aset dan persetujuan permintaan aset. Hasil *prototype* pertama dapat langsung diverifikasi pengguna sehingga pengembangan lebih terkontrol. Selanjutnya pada *increment* kedua ditambahkan 5 fitur yaitu: Penerimaan aset, maintenance aset, *monitoring* status aset, *order* pembelian aset dan pengiriman aset. Penambahan fitur pada *increment* kedua akan menghasilkan produk *prototype* kedua.

Metode *Incremental* sangat fleksibel pada pengembangan aplikasi manajemen aset “MyAsset” kedepan. Penambahan kebutuhan baru dapat ditambahkan sebagai *increment* tahap berikutnya sehingga pengembangan aplikasi sangat dinamis. Selama siklus pengembangan aplikasi yang baru pengguna masih dapat mengoperasikan aplikasi sebelumnya. Dan ketika *increment* pengembangan

aplikasi selesai pengguna mendapatkan fitur-fitur baru yang sudah tergabung dengan fitur aplikasi sebelumnya. Metode pengembangan incremental mengurangi resiko kegagalan pengembangan aplikasi karena setiap increment pengembangan terlibat melakukan verifikasi.

Referensi:

- [1] Sudrajat, Irwan, 2007. Lifecycle Asset Management, <http://assetmanagement.wordpress.com/2007/06/14/lifecycle-asset-management/>
- [2] Developer Android. (2011, December) Developer Android. [Online]. <http://developer.android.com/guide/topics/fundamentals/activities.html>
- [3] Hariyady, 2017, Pembangunan Aplikasi Mobile “Green and Clean UMM” menggunakan metode User Centered Design., Sentra Nasional Teknologi dan Rekayasa (SENTRA) 2017
- [4] Hariyady, 2015, Pembangunan Aplikasi Foto Overlay Taman Rekreasi Sengkaling menggunakan Augmented Reality, Sentra Nasional Teknologi dan Rekayasa (SENTRA) 2015
- [5] Hariyady, 2015, Pembagunan Framework Aplikasi Mobile Electronic Commerce (m-commerce),. Sentra Nasional Teknologi dan Rekayasa (SENTRA) 2015
- [6] Jogiyanto H.M., 1995. Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis. Andi Offset, Yogyakarta.
- [7] Mitchell, John S., 2006. Physical Asset Management Handbook, CLARION Technical, Boston.
- [8] Pressman, Roger : —Software Engineering: A Practitioner's Approach”, Edisi ke 6, New York : McGraw-Hill, 2005.

PEMBANGUNAN APLIKASI MANAJEMEN ASET

ORIGINALITY REPORT

7%

SIMILARITY INDEX

6%

INTERNET SOURCES

0%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Universitas Islam Indonesia

Student Paper

3%

2

www.gunadarma.ac.id

Internet Source

2%

3

Submitted to Universitas Brawijaya

Student Paper

2%

Exclude quotes Off

Exclude bibliography On

Exclude matches < 2%